

# Unsafe usage of high privileged accounts

---

Authenticating with a high privileged account

## Background

---

## Pre-requisites

---

## Risks

---

## How to check for

---

To collect sessions we can run Sharphound in a loop to gather new sessions.

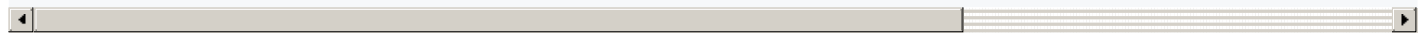
```
Collect new sessions every 10 minutes for 5 hours.  
invoke-bloodhound -CollectionMethod session -Loop -LoopInterval 00:10:00 -LoopDuration 05:00:00
```

After importing the Sharphound results to bloodhound we can run the following cypher-query to see where admin-users have sessions.

```
MATCH p=(c:Computer)-[r:HasSession]->(n:User {admincount : true}) RETURN p  
MATCH (c:Computer)-[r:HasSession]->(n:User {admincount : true}) RETURN n
```


The result can be returned as a CSV-file.

```
echo "MATCH p=(c:Computer)-[r:HasSession]->(n:User {admincount : true}) RETURN n.name,c.name;" | /usr/share/neo4j/bin/cypher-shell -
```




If you don't want to single out which users have used their admin-accounts in an insecure way you can just remove that.

```
echo "MATCH p=(c:Computer)-[r:HasSession]->(n:User {admincount : true}) RETURN n.name,c.name;" | /usr/share/neo4j/bin/cypher-shell -
```



If you are using `latex-utils` you can do the following:

```
echo "MATCH p=(c:Computer)-[r:HasSession]->(n:User {admincount : true}) RETURN n.name,c.name;" | /usr/share/neo4j/bin/cypher-shell -
```



## How to exploit

---

## Recommendation

---

## Related Vulnerabilities

---

- Insecure Usage of high privileged accounts

## References

---